

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior version, and listings, of claims in the application:

Listing of Claims:

1-20 (Canceled)

21. (Currently Amended) A hooks module for use in a computer graphics system comprising graphics hardware controlled by a graphics library having graphics library functions responsive to function calls generated by a graphics diagnostic tool executing on the computer graphics system, wherein the hooks module is configured to link ~~attach~~ the graphics diagnostic tool to at least one selected portion of the computer graphics system during normal operations of a currently-executing graphics application and without interruption to the normal operations of the graphics application.

22. (Previously Presented) The hooks module of claim 21, wherein the hooks module comprises one or more event generators, operatively located in the graphics library, configured to perform a diagnostic operation during normal operations of the graphics application.

23. (Previously Presented) The hooks module of claim 22, wherein the graphics library further includes an applications program interface (API) for invoking the graphics library functions in response to commands generated by the graphics application, and wherein the one or more event generators comprises:

at least one API event generator operatively located in the graphics library API, each configured to perform selected diagnostic operations related to an associated one or more graphics library functions.

24. (Previously Presented) The hooks module of claim 21, wherein the hooks module further comprises:

a normal operations dispatch table including function pointers to the graphics library functions;

a hooks dispatch table including function pointers to the at least one API event generator; and

a dispatch table manager constructed and arranged to copy selected portions of the normal operations dispatch table and the hooks dispatch table to an active dispatch table in the graphics library, wherein, in response to graphics application function calls, the API calls one of either the at least one API event generator or the graphics library functions in accordance with a function pointer located in the active dispatch table.

25. (Previously Presented) The hooks module of claim 22, wherein the one or more event generators comprises:

one or more internal event generators adapted to be embedded in the graphics diagnostic library to perform selected diagnostic operations in response to a request by the graphics tool.

26. (Currently Amended) The hooks module of claim 25, wherein the graphics library further includes graphics hardware control modules for controlling the graphics hardware, and

wherein at least one of the one or more internal event generators is embedded in the graphics hardware control modules to provide the graphics diagnostic tool access to an associated operation in the graphics hardware control modules after the graphics diagnostic tool is linked ~~attached~~ to the computer graphics system by the hooks module.

27. (Currently Amended) The hooks module of claim 25, wherein the graphics library includes pipeline control modules for managing a graphics pipeline in the graphics system, wherein the one or more internal event generators comprise at least one pipeline control module event generator embedded in the pipeline control modules for providing the graphics diagnostic tool access to the pipeline control modules after the graphics diagnostic tool is linked ~~attached~~ to the computer graphics system by the hooks module.

28. (Currently Amended) The hooks module of claim 25, wherein the graphics library includes device-specific control modules each configured to manage a specific portion of the graphics hardware, wherein the one or more internal event generators comprise: device-specific event generators for providing the linked ~~attached~~ graphics diagnostic tool access to the device-specific control modules.

29. (Previously Presented) The hooks module of claim 21, wherein the graphics diagnostic tool and the hooks module communicate with each other through a interprocess communications (IPC) mechanisms providing socket communications between the graphics diagnostic tool and the hooks module.

30. (Previously Presented) The hooks module of claim 21, wherein the hooks module comprises at least one first event generator that causes the graphics system to incur significant performance penalties while performing the diagnostic operations, and at least one second event generator that does not cause the graphics system to incur significant performance penalties while performing the diagnostic operations, wherein the hooks module dynamically enables the at least one first event generator to temporarily perform the diagnostic operations only when required, and to permanently enable the at least one second event generator to perform the diagnostic operations continually.

31. (Previously Presented) A method for providing a graphics diagnostic tool access to a computer graphics system having graphics hardware and a graphics library for controlling the graphics hardware in response to function calls received from a graphics application executing thereon, the method comprising:

determining dynamically during execution of the graphics application and without interrupting the execution of the graphics application whether to install an entry point event generator in the graphics library; and

installing the event generator in the graphics library while the graphics application is executing, wherein the event generator is thereafter responsive to a graphics application function call.

32. (Previously Presented) The method of claim 31, wherein installing the event generator comprises:

receiving a graphics library function call issued by the graphics application to invoke a graphics library function;

invoking the event generator associated with that graphics library function;

performing diagnostic operations by the event generator; and

calling the associated graphics library function identified in the function call.

33. (Previously Presented) The method of claim 31, wherein the graphics library is an OpenGL graphics library.

34. (Previously Presented) The method of claim 32, wherein installing the event generator comprises:

forwarding results of the selected operations to the graphics tool.

35. (Previously Presented) The method of claim 32, wherein invoking the event generator comprises:

- receiving a request from the graphics tool to perform the diagnostic operation;
- installing an active dispatch table in the graphics library wherein the active dispatch table include only function pointers to those API event generators that perform diagnostic operations that will provide results requested by the graphics tool; and
- calling one of either the API event generators or the graphics library function in response to a graphics library function call.

36. (Previously Presented) The method of claim 35, wherein installing an active discharge table comprises:

- providing a normal operations dispatch table having function pointers to the graphics library functions;
- providing a hooks dispatch table including function pointers to the API event generators; and
- copying selected portions of the normal operations dispatch table and the hooks dispatch table to the graphics library to form the active dispatch table in the graphics library.

37. (Previously Presented) The method of claim 31, further comprising:

- embedding in the graphics library an internal event generator; and
- enabling the internal event generators in the graphics library in response to a request from the graphics tool; and
- performing, by the internal event generator in response to the graphics tool, selected diagnostic operations, wherein the results provided to the graphics tool includes information pertaining to the graphics system.

38. (Currently Amended) A computer graphics system comprising:

- a hooks module integrated within a computer graphics system for dynamically linking ~~attaching~~ a graphics diagnostic tool to at least one selected portion of the computer graphics system while a graphics application is executing on the graphics system, and without interrupting the execution of the graphics application.

39. (Previously Presented) The computer graphics system of claim 38, wherein the hooks module comprises:

a plurality of internal event generators embedded in a graphics library of the computer graphics system, each said event generator constructed and arranged to perform diagnostic operations during normal operations of the graphics application in response to requests generated by the graphics diagnostic tool.

40. (Previously Presented) The computer graphics system of claim 38, wherein the hooks module comprises at least one internal event generator that provides the graphics tool with access to internal state and control flow of the graphics system, and wherein the at least one internal event generators generate a stream of state information sufficient to recreate a current state of the graphics system.

41. (Previously Presented) The hooks module of claim 23, wherein each of the at least one API event generator is configured to be dynamically installed in the graphics to perform the selected diagnostic operations.

42. (Previously Presented) A system for providing a graphics diagnostic tool access to a computer graphics system comprising graphics hardware and a graphics library for controlling the graphics hardware, the graphics library including graphics library functions responsive to function calls generated by an executing graphics application, the system comprising:

a plurality of internal event generators embedded in the graphics library to perform diagnostic operations on the computer graphics system during normal operations of the graphics application.

43. (Previously Presented) The system of claim 42, wherein at least one of the diagnostic operations performed by the plurality of internal event generators provides the graphics tool with information pertaining to internal state and control flow of the computer graphics system.

44. (Previously Presented) The system of claim 42, wherein the graphics library further comprises at least one graphics hardware control module for controlling the graphics hardware, and wherein at least one of the plurality of internal event generators is embedded in one or more of the at least one control module to provide the graphics diagnostic tool access to an associated operation in the one or more control modules when the graphics diagnostic tool is attached to the computer graphics system.

45. (Previously Presented) The system of claim 42, wherein the graphics library comprises at least one pipeline control module for managing at least a portion of a graphics pipeline of the graphics system, and wherein the plurality of internal event generators comprise:

at least one pipeline control module event generator each embedded in a pipeline control module for providing the graphics diagnostic tool access to that pipeline control module when the graphics diagnostic tool is attached to the computer graphics system.

46. (Previously Presented) The system of claim 42, wherein the graphics library comprises at least one device-specific control module for managing at least a portion of the graphics hardware, and wherein the plurality of internal event generators comprises:

at least one device-specific event generator each providing the graphics tool access to one of the at least one device-specific control module when the graphics diagnostic tool is attached to the computer graphics system.